SAND REPORT

SAND2003-0080
Unlimited Release
Printed January 2003

Computational Algorithms for Device-Circuit Coupling

Eric R. Keiter, Scott A. Hutchinson, Robert J. Hoekstra, Eric L. Rankin, Thomas V. Russo, and Lon J. Waters

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/ordering.htm

SAND2003-0080
Unlimited Release
Printed January 2003

Computational Algorithms for Device-Circuit Coupling

Eric R. Keiter, Scott A. Hutchinson, Robert J. Hoekstra and Eric L. Rankin
Computational Sciences

Thomas V. Russo and Lon J. Waters
Component Information and Models

Sandia National Laboratories
P.O. Box 5800
Mail Stop 0316
Albuquerque, NM 87185-0316

Abstract

Circuit simulation tools (e.g., **SPICE**) have become invaluable in the development and design of electronic circuits. Similarly, device-scale simulation tools (e.g., **DaVinci**) are commonly used in the design of individual semiconductor components. Some problems, such as single-event upset (SEU), require the fidelity of a mesh-based device simulator but are only meaningful when dynamically coupled with an external circuit. For such problems a mixed-level simulator is desirable, but the two types of simulation generally have different (sometimes conflicting) numerical requirements. To address these considerations, we have investigated variations of the two-level Newton algorithm, which preserves tight coupling between the circuit and the PDE device, while optimizing the numerics for both.

This page is left intentionally blank

Contents

This page is left intentionally blank

Figures

Tables

This page is left intentionally blank

1 Introduction

This report contains results of an investigation into mixed-level electrical simulation algorithms. The focus has been on coupling techniques for electrical simulations of different fidelity: circuit simulation [2, 12] and device simulation [3]. Analog circuit simulation has been used extensively in the electronic design community and is based upon describing an electrical circuit as a system of network-coupled differential algebraic equations (DAE's). Device simulation is a higher fidelity simulation, which describes a single semiconductor device with a set of coupled partial differential equations (PDE's), discretized on a spatial mesh. These different types of simulation generally present different solver (time integration, nonlinear and linear) requirements, so coupling them together is potentially problematic. The two-level Newton algorithm address these issues by isolating the circuit and PDE device phases of the calculation. Variations of this method have been the focus of this work.

1.1 State of the Field

Coupling algorithms between PDE device simulation and analog circuit simulation have been studied over the past 20 years [10, 20, 24, 17, 30]. Most of these studies employed a full-Newton approach, which is still the most commonly applied method for commercial simulators.

In the early 1990's, a new approach to mixed-level simulation was published by Mayaram [15, 16]. Mayaram proposed the two-level Newton algorithm, and successfully applied it to mixed-level simulation. Later, Rotella [25] also used the two-level approach, building on the work of Mayaram. Unlike earlier work, which was applied to just DC operating point (steady state) and transient calculations, both Mayaram and Rotella also applied their approach to small-signal AC analysis. Both Mayaram and Rotella reported that the two-level Newton algorithm was much more robust than the full-Newton for mixed-level simulation, although it tended to be slower.

1.2 Accomplishments

This project has required substantial code development, which has progressed in three stages: (1) the initial development of **Xyce**, a massively parallel circuit simulator, which also served as the code framework for this work, (2) the development of PDE devices within the **Xyce** framework, and (3), the development of various nonlinear coupling algorithms within the framework.

As noted, the simulator framework used in this work has been the **Xyce** Parallel Electronic Simulator whose development began in 1999. **Xyce** is an object-oriented, massively parallel simulator written in C++, and was designed to be closely compatible with Spice3f5 [23], a circuit simulator which is the defacto industry standard. In developing **Xyce**, great advantage was taken of the various solver and partitioning algorithm libraries developed at Sandia National Laboratories. As such, the code incorporates the **Trilinos** [5], **NOX**, **Zoltan** and **Chaco** [11] libraries. In developing this coupled capability, it was necessary for **Xyce** to achieve some level of maturity prior to attempting PDE device simulation, so the early part of the project mostly focused on core **Xyce** development.

By the latter half of this project, **Xyce** was robust enough to accommodate PDE based device models. Under the "core" development mentioned above, the device package of **Xyce** was designed with a highly object-oriented structure, and within that structure every device (analog circuit device as well as PDE device) sits behind the same base class interface. This design allowed the code to interface with the PDE-based devices just as it would with other device-model types (e.g., analog, behavioral, etc.). In the project, two PDE devices were developed - a one-dimensional device and a two-dimensional device. Both devices are based on a drift-diffusion model [14, 28, 32], in which Poisson's equation, as well as electron and hole continuity equations are solved at every point on a spatial mesh. Both devices relied on the Scharfetter-Gummel discretization [27], and as such were able to utilize many of the same base class flux evaluation functions.

For the final stage of this project, a number of different nonlinear solver coupling algorithms were built into the code. These include the full-Newton, two-level-Newton, modified two-level-Newton, and two-level-Newton with continuation. As with the device pacakge, the **Xyce** nonlinear solver package was designed with an object-oriented structure, and each of these two-level solver options sits behind the same base-class interface. As a result, the choice of nonlinear solver is invoked is mostly opaque to the rest of the code. Based on our studies to date, the two-level-Newton with continuation is the most robust general approach of the four.

2 Problem Definition

This section defines the general circuit problem, as well as the general device problem, in terms of the set of equations to be solved. First is a brief description of the circuit problem, then a detailed description of the device problem. Finally coupling equations for mixed-level simulation are presented.

2.1 Circuit Simulation

Methods of analog circuit simulation (e.g., SPICE [22]) are well described by Vlach and Singal [29], among others [6, 8, 31]. The circuit to be simulated is represented as a system of coupled DAE's, which are obtained from Kirchoff's current and voltage laws (KCL and KVL, respectively). In general, there are several mathematical formulations that may be used but in practice, nearly all circuit simulators use the "modified KCL" formulation [26]. This is the formulation used by **Xyce**, and has been described in detail in the **Xyce** math document [13].

2.2 Device Simulation

Methods of device simulation are described by many references including Kramer [14] and Selberherr [28]. As with circuit simulation, there are several different approaches that may be used. However, the most common one, and the one that is used for this work, is the drift-diffusion (DD) formulation. This formulation consists of three coupled PDE's: a single Poisson equation for electrostatic potential and two continuity equations; one each for electrons and holes.

Poisson equation

The electrostatic potential ϕ satisfies Poisson's equation:

$$-\nabla \cdot (\epsilon \nabla \phi(x)) = \rho(x) \tag{1}$$

where ρ is the charge density and ϵ is the permittivity of the material. For semiconductor devices, the charge density is determined by the local carrier densities and the local doping,

$$\rho(x) = q(p(x) - n(x) + C(x)) \tag{2}$$

Here $p(x)$ is the spatially-dependent concentration of holes, $n(x)$ the concentration of electrons, and q the magnitude of the charge on an electron. $C(x)$ is the total doping concentration, which can also be represented as $C(x) = N_D^+(x) - N_A^-(x)$, where N_D^+ the concentration of positively ionized donors, N_A^- the concentration of negatively ionized acceptors.

Species continuity equations

The continuity equations relate the convective derivative of the species concentrations to the creation and destruction of particles ("recombination/generation").

$$\frac{\partial n(x)}{\partial t} + \nabla \cdot \Gamma_n = -R(x) \quad (3)$$

$$\frac{\partial p(x)}{\partial t} + \nabla \cdot \Gamma_p = -R(x) \quad (4)$$

Here n is the electron concentration and p is the hole concentration. R is the recombination rate for both species. Γ_n and Γ_p are particle fluxes for electrons and holes, respectively. The sign of R is chosen because R is usually expressed as a recombination rate, and is positive if particles are annihilating. The right hand sides are equal since creation and destruction of carriers occurs in pairs.

One way in which the drift-diffusion model differs from other common formulations is the manner in which the quantities Γ_n and Γ_p are determined. The expressions used are:

$$\Gamma_n = n(x)\mu_n E(x) + D_n \nabla n(x) \quad (5)$$

$$\Gamma_p = p(x)\mu_p E(x) + D_p \nabla p(x) \quad (6)$$

Here μ_n, μ_p are mobilities for electrons and holes, and D_n, D_p are diffusion constants. $E(x)$ is the electric field, which is given by the gradient of the potential, or $-\partial\phi/\partial x$.

Discretization

For this work, a box integration scheme has been used to discretize the PDE's. A complete description of this approach is given in several places, such as Kramer [14]. A brief description is given here. The main advantage of using box integration is that it allows one to use an unstructured mesh to discretize the domain. An example of such a mesh for a MOSFET device is shown in Figure 1. The mesh in the figure was generated using the SGFramework [14] mesh generator. It is generally desirable to increase mesh refinement in regions where the solution is likely to show greater variation in space. For most semiconductor devices, this will be in regions where the doping profile abruptly changes.

Each of the three differential equations contains a divergence operator as its main differential operator. The general form that is followed by all three equations is:

$$\nabla \cdot \Gamma = S \quad (7)$$

where Γ is the flux (which may be electric field or charged particle flux) and S is a source term.

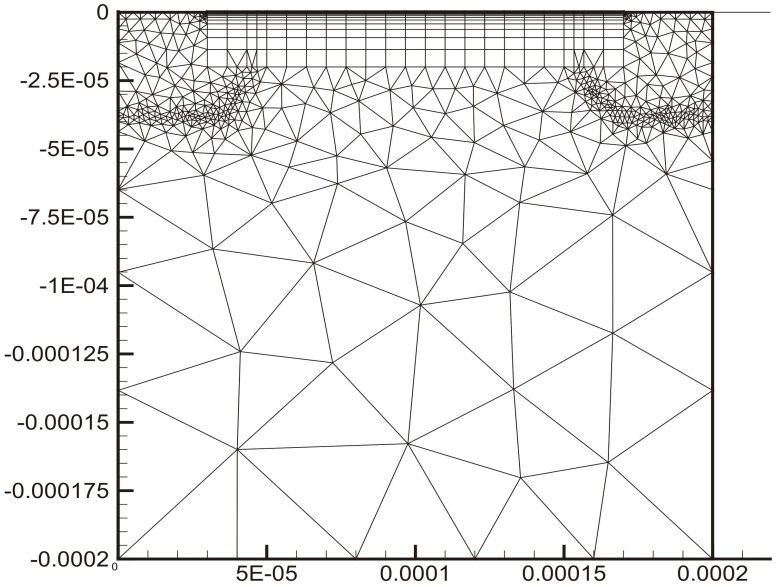

Figure 1. MOSFET Mesh.

The box integration method approximates an equation of the form of 7 by constructing an abstract volume, or box around each node on a mesh. An example is shown in Figure 2, in which the box around node 1 is represented by the area in gray. Each mesh node (such as node 1 in the figure) is connected by edges to a discrete number of neighbors (nodes 2-7 in the figure). The box is the area contained by the perpendicular bisectors of the connecting edges. Equation 7 is integrated over the box, giving:

$$\int\int_A (\nabla \cdot \Gamma) da = \int\int_A S da \tag{8}$$

Green's theorem can then be applied to the left side of this expression, converting the area integral into a line integral:

$$\oint_{\partial A} (\Gamma \cdot n) dl = \int\int_A S da \tag{9}$$

The discrete form, based on the box around node 1 is:

$$\sum_{i=2}^{7} \Gamma_i L_{1i} = S_1 A_1 \tag{10}$$

where L_{1i} is length of the perpendicular bisector between node 1 and node i and A_1 is the area of the integration box around node 1. Equation 10 is the general

algebraic form that is used to discretize Poisson's equation and the two continuity equations. In the case of Poisson's equation, Γ_i becomes the electric field perpendicular to face i, and S_1 becomes the charge density term. For the continuity equations Γ_i is the flux of electrons or holes, and S_1 is a generation-recombination term.

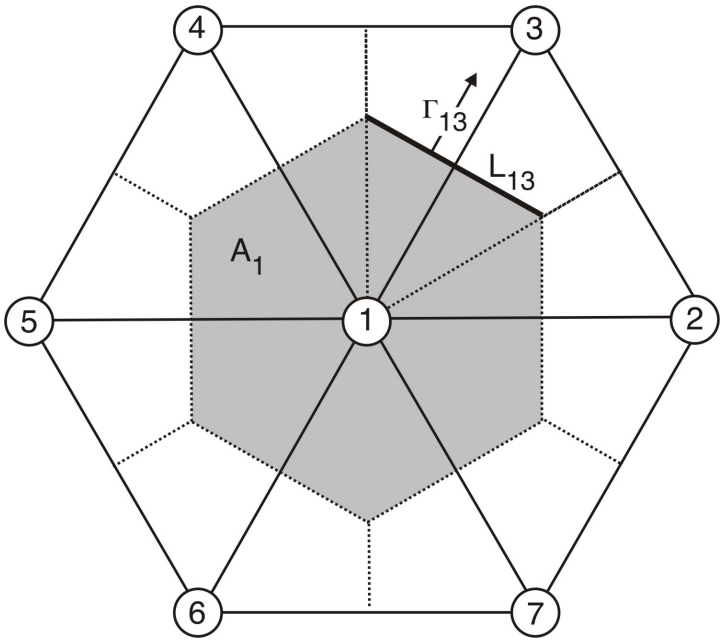

Figure 2. Box Integration.

Stability

The solution of the set of semiconductor transport equations is notoriously difficult. This difficulty comes mainly from the fact that the nature of electron and hole transport varies widely over the domain. That is, far from junctions, charged species transport is almost completely diffusional, while in the near case it becomes highly convective. As a result, a numerical scheme which is optimal in the junction will often be inappropriate elsewhere and vice-versa. Traditional discretization schemes generally result in spurious oscillations near the depletion region, if they converge at all.

The most popular technique for avoiding oscillations is one originally proposed by Scharfetter and Gummel [27], referred to in this document as the SG method. The SG method can be viewed as a dynamic up-winding scheme. In the limit of zero electric field (purely diffusional transport) it reduces to a standard central

difference, while in the limit of infinite electric field (purely convective transport) it reduces to a difference which is strictly upwind.

The derivation of SG is based on the one dimensional case and is given in detail by both Selberherr [28] and Kramer [14]. It starts with a simplified drift-diffusion equation for charged species flux:

$$\Gamma_n = \mu_n \left(n(x)E(x) + \frac{d}{dx} n(x) \right) \tag{11}$$

The simplification used is that the mobility μ_n, and the diffusion constant D_n, are the same quantity. This is valid as long as the Einstein relation is valid and the variables are scaled correctly. Applying this equation between two mesh nodes results in expressions for Γ_n and $n(x)$ at the midpoint between the two nodes. To obtain the midpoint values, the values of $\phi(x)$ and $n(x)$ must be known at both mesh nodes, and equation 11 is solved using an integration factor. After a performing the algebra (see [14] and [28] for details), the following expressions are obtained for the midpoint electron flux:

$$\text{aux1(x)} = \frac{x}{\text{sinh}(x)} \tag{12}$$

$$\text{aux2(x)} = \frac{1}{1 + \text{exp}(x)} \tag{13}$$

$$n|_{\text{mdpt}} = n_i \text{aux2} \left(\frac{\phi_i - \phi_j}{2U_t} \right) + n_j \text{aux2} \left(\frac{\phi_j - \phi_i}{2U_t} \right) \tag{14}$$

$$\frac{dn}{dx}|_{\text{mdpt}} = \text{aux1} \left(\frac{\phi_i - \phi_j}{2U_t} \right) \frac{n_j - n_i}{h_{i,j}} \tag{15}$$

$$\Gamma_n|_{\text{mdpt}} = \mu_n|_{\text{mdpt}} \left(n|_{\text{mdpt}} \frac{\phi_i - \phi_j}{h_{i,j}} \frac{dn}{dx}|_{\text{mdpt}} \right) \tag{16}$$

An analogous set of flux expressions can be obtained for holes as well. These expressions are very convenient for the box integration method, as they provide both electron and hole fluxes at the midpoints between mesh nodes, exactly as required by the scheme.

Unfortunately the SG method has some drawbacks. Like many stabilization schemes, it introduces artificial diffusion into the problem, and for problems of more than one

dimension crosswind diffusion can be an issue. However, it is used by most of the popular commercial device simulators and for historical reasons is somewhat of an industry standard. As such, the SG method was used for stabilization in this work.

Initial Condition

Equilibrium Approximation

Even when the SG method is employed, PDE semiconductor simulations will often fail to converge if not given a good initial guess. A simple initial guess, based on the equilibrium electron and hole concentrations can be calculated using information about the doping profile.

$$\text{if } C(x) >= 0 \quad n(x) = \frac{1}{2}\left(\sqrt{C(x)^2 + 4Ni^2} + |C(x)|\right) \tag{17}$$

$$\text{if } C(x) < 0 \quad n(x) = \frac{2Ni^2}{\left(\sqrt{C(x)^2 + 4Ni^2} + |C(x)|\right)} \tag{18}$$

(19)

$$\text{if } C(x) <= 0 \quad p(x) = \frac{1}{2}\left(\sqrt{C(x)^2 + 4Ni^2} + |C(x)|\right) \tag{20}$$

$$\text{if } C(x) > 0 \quad p(x) = \frac{2Ni^2}{\left(\sqrt{C(x)^2 + 4Ni^2} + |C(x)|\right)} \tag{21}$$

(22)

where $C(x)$ is the spatially dependent doping concentration, and Ni is the intrinsic carrier concentration. Once the electron and hole concentrations have been calculated, the equilibrium potential can be approximated by:

$$\phi(x) = Vt \ln\left(N_D/p(x)\right) \tag{23}$$

Vt is the thermal voltage, or k_bT/q. N_D in this case is a scalar quantity, representing the maximum donor concentration across the spatial domain.

Nonlinear Poisson

For harder problems, a better initial guess than can be obtained from the above expressions is often necessary. Semiconductor PDE problems tend to be more

difficult to solve for high doping levels and also for complex geometries. One can approximate the solution to the drift diffusion semiconductor problem by solving a nonlinear Poisson equation, given by:

$$-\nabla \cdot (\epsilon \nabla \phi(x)) = q\left(N_A e^{((\phi_{min}-\phi(x))/Vt)} - N_D e^{((\phi(x)-\phi_{max})/Vt)} + C(x)\right) \quad (24)$$

This equation is the same as equation 1, except that now the electron and hole densities are functions of the electrostatic potential, so the equation is nonlinear. The variables N_A and N_D are the maximum acceptor and donor concentrations. In equilibrium, they will approximate the maximum hole and electron concentrations. This equation can easily be solved on the same mesh as the full drift-diffusion formulation using a damped Newton method. Upon solution, the electron and hole densities are obtained by simply evaluating the density terms in equation 24:

$$p(x) = N_A e^{((\phi_{min}-\phi(x))/Vt)} \quad (25)$$

$$n(x) = N_D e^{((\phi(x)-\phi_{max})/Vt)} \quad (26)$$

For the case of no voltage being applied across the device, solving the nonlinear Poisson gives a result that is very close to the drift-diffusion result, but is much easier to obtain. While equation 24 is nonlinear, it tends to be very well behaved from a nonlinear solver point of view. Most of the difference between the nonlinear Poisson solution and the DD solution for the unbiased case comes from the fact that equation 24 does not account for generation-recombination effects.

In running the simulation, the result of the initial condition calculation is applied before performing the initial steady-state calculation (often referred to as the DC operating point calculation). After the main calculation is underway (transient, DC sweep, etc.), the results of the nonlinear Poisson solution are no longer a valid initial guess, because equation 24 assumes equilibrium conditions.

Boundary Conditions

Boundary conditions at the electrodes for all three variables are Dirichlet, while boundary conditions everywhere else are Neumann. At non-electrode boundaries, no current should escape, so the Neumann condition is for zero flux. In the box integration scheme, this boundary condition is automatically imposed by default. The values imposed on electron and hole densities at the electrode boundaries are determined from the doping levels at those boundaries. In this work, we have applied the following expressions:

$$n_{bc} = \frac{1}{2}\left(\sqrt{C_{bc}^2 + 4Ni^2} + C_{bc}\right) \tag{27}$$

$$p_{bc} = \frac{1}{2}\left(\sqrt{C_{bc}^2 + 4Ni^2} - C_{bc}\right) \tag{28}$$

where n_{bc} is the imposed value of electron density at a boundary, while p_{bc} is the imposed value for hole density. Also, C_{bc} is the doping level at the boundary. These expressions were taken from Selberherr [28], and enforce both thermal equilibrium and a vanishing space charge at ohmic contacts.

The boundary conditions on the electrostatic potential are external inputs to the problem, which may come from the user or from an external circuit simulator. Their application, however, is not as straightforward as may first appear. Semiconductor devices will establish their own internal potential variation due to charge separation caused by the doping. This can be observed by considering equation 23. Doping levels will usually be different at each electrode boundary, so equation 23 will evaluate to a different value for each one.

Therefore, when an external potential is applied to a PDE device electrode, the boundary condition applied at that electrode has to include an offset due to the built in potential. This offset is determined by evaluating equation 23 at each boundary. The potential boundary condition which is applied is then:

$$\phi_{bc} = \phi_{ckt} + \phi_{offset} \tag{29}$$

It should be noted that for some PDE devices, the doping will not be constant along an electrode boundary. For such cases, the imposed boundary conditions on all three variables need to vary accordingly along the electrode.

2.3 Coupling Equations

A mixed level problem will include the full set of circuit equations as well as the full set of device equations. In addition, such a simulation will also require extra equations associated with the coupling to complete the system. In order to couple a PDE level device simulation to a circuit simulation, it is necessary for the PDE device simulator to fulfill the role of a SPICE-style analog device. While the **Xyce** device-package interface makes this relatively easy from a programming standpoint, in practice, the PDE-device must provide currents and conductances as a function of voltage.

The interface between a PDE device simulation and a circuit simulation occurs at the device electrodes, each of which are physically connected to a node of

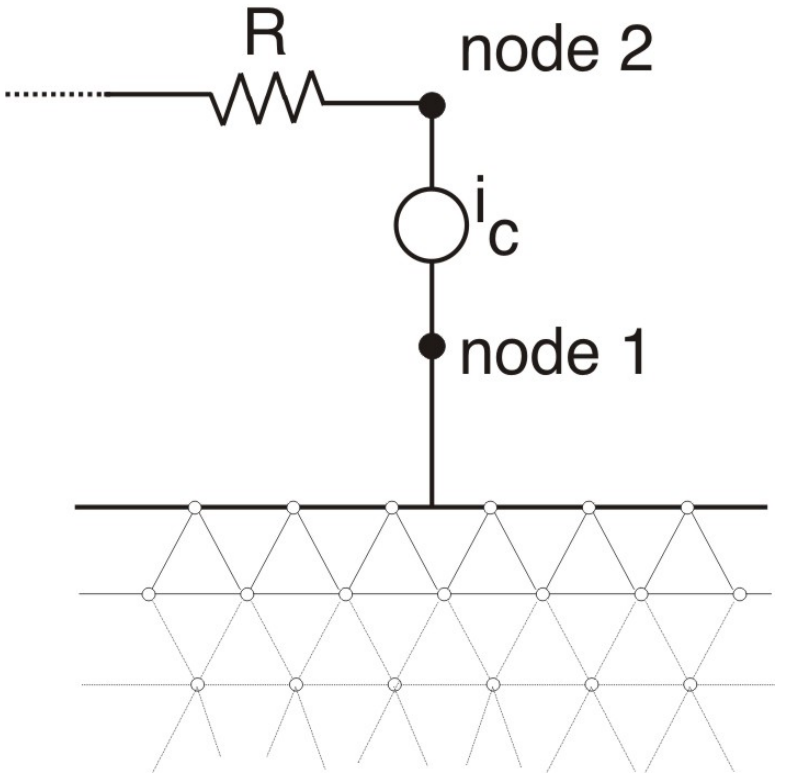

Figure 3. Circuit and PDE device mesh connected together.

the circuit. A typical connection between a circuit node and a device electrode is illustrated in Figure 3. The current flowing between the device and the circuit must satisfy the following equation:

$$\sum_{i=1}^{N} A_i \cdot (J_i^n + J_i^p) = i_c \tag{30}$$

N is the number of mesh nodes along the electrode, A_i is the surface area associated with mesh node i. J_i^n and J_i^p are the electron and hole current densities, respectively. The term on the right hand side of equation 30, i_c, is the circuit current which flows into circuit node 1.

Potentially, a large number of nodes on the PDE mesh lie on an electrode boundary, so the summation necessary to obtain the total current may involve a relatively large number of terms. For a three dimensional simulation, N could potentially be in the thousands.

Equation 30 is essentially a KCL equation, so while it involves terms from the PDE

device, it is natural to think of it as belonging to the circuit part of the problem. The coupling between the device and the circuit is also enforced on the PDE side of the problem through boundary conditions on the electrode potentials. Semiconductor devices, even when unbiased, possess a certain amount of internal potential variation, due to variations in the doping. A semiconductor device which is not connected to a circuit, and has no external potentials applied to it, will still have this internal potential variation. As a result, any potentials applied to a PDE device that have been obtained from a connected circuit must be adjusted, to account for the internal variation. This results in the equation:

$$\phi_{bc} = \phi_{circuit} + \phi_{offset} \tag{31}$$

Here ϕ_{bc} is the voltage boundary condition to be applied to the PDE device on the electrode. $\phi_{circuit}$ is the voltage at the circuit node which is connected to the electrode. ϕ_{offset} is different for each electrode and depends upon the local doping. It is common for the variables of the PDE simulation to be scaled differently than the variables of the circuit equation. Thus, both equation 30 and 31 need to take this into account.

3 Solution Methods

This section focuses on solver methods employed by circuit, device and mixed-level simulation. Circuit and device simulation have a number of similarities, without which a two-level Newton approach would not be applicable. First, the similarities between circuit and device simulation from a solver point of view are discussed, followed by a discussion of some of their differences. A discussion of coupling algorithms is the last part of this section.

3.1 Similarities

Circuit simulation and PDE device simulation are similar in several regards. Both deal with the same basic physical quantities - electrostatic potential and current. Both problems are nonlinear and implicit in time.

An illustration of the solver structure followed by both styles of simulation is shown in Figure 4. This structure consists of a nested set of solver loops. The outer loop is a control loop, which may be a time integration loop or a DC sweep. The middle level is a nonlinear solver, and the inner-most entity is the linear solver.

The system of equations at each time step is represented as

$$\mathbf{f}(\mathbf{w}) = 0 \tag{32}$$

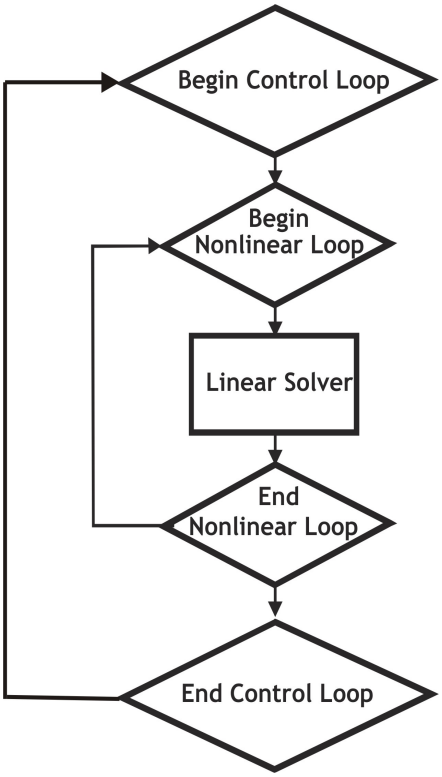

Figure 4. Solver flowchart.

in which $\mathbf{w}$ is the solution vector and $\mathbf{f}$ is the system of equations. Equation 32 is solved using a Newton method, in which the solution $\mathbf{w}$ is obtained from repeated solves of the following equations:

$$\mathbf{J}\Delta\mathbf{w} = -\mathbf{f} \tag{33}$$

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \Delta\mathbf{w} \tag{34}$$

k is the Newton iteration index. This set of equations is repeatedly evaluated until Equation 32 is satisfied.

3.2 Differences

Circuit simulation and device simulation are different enough to have somewhat different solver needs, for the time integrator, nonlinear solver, and linear solver. For the linear solver phase, PDE device simulation is usually successful with iterative

methods. For circuit simulation direct methods are generally much more reliable than iterative methods, although research in this area is ongoing [7].

For the nonlinear solve, device simulation can easily be handled using a conventional damped Newton, as long as the simulator is provided with a good initial guess. Equation 34 is modified as:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha\Delta\mathbf{w} \tag{35}$$

in which α is a scalar that is adjusted such that $\mathbf{w}_{k+1}$ results in a reduced value for $|\mathbf{f}(\mathbf{w}_{k+1})|$.

Circuit simulation is somewhat more problematic, in that applying a single scalar α to the entire update is often not adequate to converge the operating point calculation. Many of the nonlinear circuits in the **Xyce** test suite will fail if such an approach is taken. A more effective approach for circuits is voltage limiting [13]. The junction voltages of nonlinear devices such as BJT's, MOSFET's and diodes are restricted from varying too much from one Newton iteration to the next. This is similar to damping, except that it is applied on a device-by-device basis, and the limiting applied in one device may not be consistent with the limiting applied in another. The implementation in **Xyce** is similar to that of legacy circuit simulators, in that a correction due to voltage limiting has to be applied on the right hand side of equation 33. This takes the following form:

$$\mathbf{J}(\mathbf{w}')\Delta\mathbf{w}_{total} = -\mathbf{f}(\mathbf{w}') + \mathbf{J}(\mathbf{w}')\Delta\mathbf{w}_{limit} \tag{36}$$

where $\Delta\mathbf{w}_{limit}$ is the change in $\mathbf{w}$ at the current Newton step due to voltage limiting, and $\Delta\mathbf{w}_{total} = \Delta\mathbf{w}_{Newton} + \Delta\mathbf{w}_{limit}$. The quantities $\mathbf{w}_{k+1}$, $\mathbf{w}_k$ and $\mathbf{w}'_k$ are solution vectors related by:

$$\mathbf{w}'_k = \mathbf{w}_k + \Delta\mathbf{w}_{limit} \tag{37}$$

$$\mathbf{w}_{k+1} = \mathbf{w}'_k + \Delta\mathbf{w}_{Newton} \tag{38}$$

The limits are generally evaluated and applied in terms of voltage drops within individual devices, so the correction term in equation 36 is applied during the load phase of the individual devices.

Although voltage limiting has been around for a long time [21], and has been suprisingly successful, it has the drawback of being incompatible with other nonlinear

solver methods. It is a source of non-smooth behavior in part of the solver, which can be problematic. More recently, homotopy methods have been applied successfully to circuit simulation. [18, 19], and they have become the method of choice for modern circuit codes. Currently, **Xyce** does not have a homotopy option for nonlinear solves, but this is a planned capability for the coming year.

3.3 Coupling Algorithms

As described in the Introduction, techniques for circuit-device coupling have been described by Mayaram [16], and later by Rotella [25]. In these works, two general approaches have been presented, the “full Newton” method and the “two-level Newton” method. These two approaches are illustrated in Figures 5, 6, and 7 respectively.

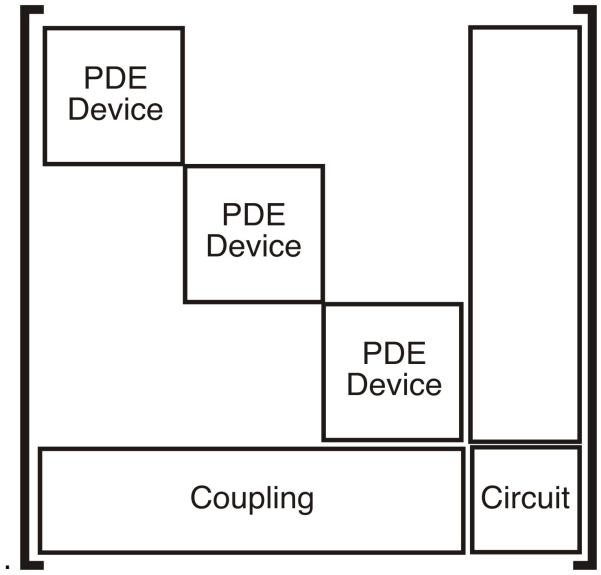

Figure 5. Full Newton Jacobian Matrix.

Tight Coupling: Full Newton

The “full Newton” method is one in which the entire problem is solved simultaneously as part of a single nonlinear system of equations. Figure 5 contains an abstract representation of the resulting Jacobian matrix. Each PDE device is represented by large sub-blocks of this matrix, and the circuit portion of the problem is also represented by a sub-block. The ordering in the diagram has all of the circuit components in the lower right corner of the matrix, but this choice was based more

on expediency than reality. In general, this will not be the initial ordering chosen by **Xyce**. In its current capability, **Xyce** will treat each PDE device the same as it would any other traditional **SPICE**-style device in the circuit from a topological perspective. As a result, the “circuit” part of the matrix will often not be represented by a single continuous sub-block, but rather will have many components in and around the PDE device sub-blocks.

Loose Coupling: Two-Level Newton

The “Two-Level Newton” was first proposed by Mayaram [16]. In this approach, the PDE device and circuit problems are considered separately, with the required information being passed back and forth between the simulators. An illustration of this approach, from the perspective of the Jacobian matrices is shown in Figure 6. The adjective “Two-Level” refers to the fact that the nonlinear solution of the circuit problem is considered as an outer level control loop, while for PDE device problem(s), the nonlinear solution is considered to be the inner level. In the most common implementation, the inner level PDE device problem is completely solved at each circuit Newton step. This results in a much slower simulation than the “full Newton” approach, but it has been shown to be much more robust [16, 25].

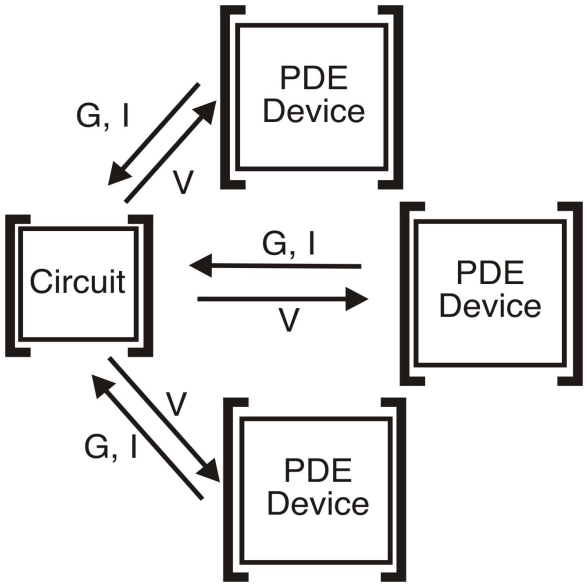

Figure 6. Two-Level Newton Jacobian Matrices.

The PDE sub-problems are solved in the same manner as a stand-alone PDE simulator, with voltages from the external circuit applied as boundary conditions. The circuit problem which comprises the outer loop, is solved similarly to a stand-alone

circuit code as well, with the PDE devices represented by extracted conductances and currents. The procedure for extracting device conductances has been described in detail by Mayaram [16]. The sub-block in the circuit matrix associated with each PDE device is an N x N conductance matrix, where N is the number of electrodes. Each entry in this conductance matrix is given by

$$G_{mn} = \frac{\partial i_m}{\partial V_n} = \frac{\partial I_m}{\partial V_n} + \frac{\partial I_m}{\partial \mathbf{w}} \frac{\partial \mathbf{w}}{\partial V_n} \tag{39}$$

m is the row index and n is the column index. From the perspective of the circuit Jacobian, m corresponds to the KCL equation associated with the circuit node attached to the mth electrode of the PDE device, and n corresponds to the voltage variable for the circuit node attached to the nth electrode. i_m is the current at the circuit node attached to electrode m. I_m is also a current, but represents the current at electrode m, from the perspective of the PDE simulation. Of course,

$$i_m(V_n) = I_m(\mathbf{w}, V_n) \tag{40}$$

Most of the terms in Equation 39 correspond to Jacobian terms that need to be calculated for the full Newton approach. The first term of Equation 39, $\partial I_m / \partial V_n$, corresponds to the PDE device contribution to the circuit node KCL on the Jacobian matrix diagonal, and is zero if $m \neq n$. The vector $\partial \mathbf{w} / \partial V_n$ is determined by first rewriting equation 32 as:

$$\mathbf{f}_{PDE}(\mathbf{w}_{PDE}(V), V) = 0 \tag{41}$$

This is the equivalent of Equation 32, except that it is only with respect to one PDE sub-problem. $\mathbf{f}_{PDE}$ is the residual vector of the PDE device, $\mathbf{w}_{PDE}$ is the solution vector for the PDE device, and V refers to externally applied voltages, which will be coming from the connected circuit. Taking the derivative of this system with respect to V, and re-arranging, the following linear system is obtained:

$$\mathbf{J}_{PDE} \frac{\partial \mathbf{w}_{PDE}}{\partial V_n} = -\frac{\partial \mathbf{f}_{PDE}}{\partial V_n} \tag{42}$$

where $\mathbf{J}_{PDE}$ is the Jacobian matrix from the PDE problem, or $\partial \mathbf{f}_{PDE} / \partial \mathbf{w}_{PDE}$ This linear system has to be solved for each electrode. $\partial \mathbf{f}_{PDE} / \partial V_n$ is a sparse vector with nonzero terms only for mesh elements connected directly to the electrode n. It corresponds to a single column of the full-Newton Jacobian.

Xyce has a slightly different implementation than the one described by Mayaram [16]. An illustration of this implementation, on the Jacobian level, is shown in Figure 7. This design was chosen in part for expedient implementation, as the same matrix structure can be used at both levels of the two-Level Newton. Equations can be removed from the problem by placing 1.0 on the diagonal and 0.0 at every other

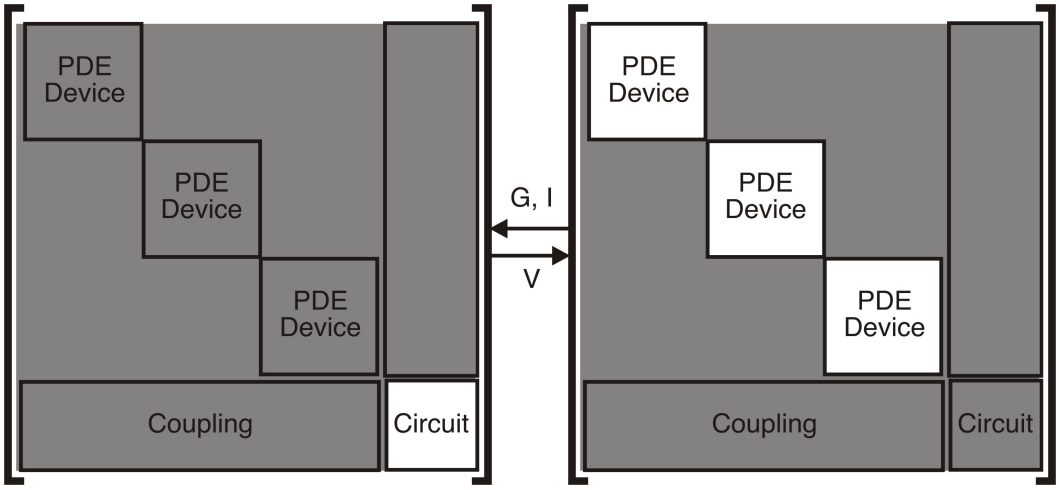

Figure 7. Two-Level Newton Jacobian Matrices, **Xyce** implementation. Shaded areas are "turned off".

location in the problem's respective matrix row, and 0.0 for the corresponding element in the residual vector. By using this shortcut, different parts of the problem can be "turned off" and "turned on", depending upon the current two-level Newton phase.

This implementation has some advantages. An optimal approach to a transient mixed-level simulation is to have the simulator apply a two-level Newton algorithm for the DC operating point, and a full Newton algorithm during the transient phase, and switching between the two modes requires very little work. During the inner loop of the two-level solve, all the PDE devices are solved simultaneously even though they are not coupled together. Several iterations of equation 42 can also solved simultaneously as well, with the nth electrode of each PDE device addressed in one matrix solve.

Loose Coupling: Modified Two-Level Newton

One of the major difficulties of solving a mixed-level problem is that PDE simulation is very sensitive to the initial guess. The voltages of connected circuit nodes provide voltage boundary conditions to PDE devices, and over the course of a two-level-Newton solve, these voltages may change a great deal at each outer loop step. A PDE simulator can often handle abrupt voltage boundary condition changes, if they are on the order the built-in potential of the device. Changes much greater than that (hundreds of volts, or thousands of volts, etc.) will almost always prevent the PDE problem from converging on the inner Newton loop, if the result of

the previous inner loop solve is used as the initial guess.

Mayaram [16] has proposed a modified two-level-Newton scheme, in which a prediction of the PDE device solution can be obtained using the derivative information of the two-level Newton. This predicted solution can then be used as an initial guess for the PDE problem's Newton loop. Mayaram uses this expression to get the initial guess for each problem:

$$\mathbf{w}_{PDE}^{k+1} = \mathbf{w}_{PDE}^{k} + \sum_{n=0}^{n=N} \left[\frac{\partial \mathbf{w}_{PDE}}{\partial V_n} \right] \Delta V_n \tag{43}$$

where k is the iteration index of the outer Newton loop. ΔV_n is the change in applied voltage at the electrode n between outer loop iterations. A potential drawback of this approach is that if ΔV_n is large enough, and negative, the resulting $\mathbf{w}^{k+1}$ may contain nonphysical negative species densities. Equation 43 has been implemented in **Xyce**, but so far has not been very successful because of this issue. However, using this technique to obtain the initial guess has been shown to be very effective [16], so more study is needed. Possibly, if voltage limiting is employed (discussed later in this document), the change in nodal voltage can be kept small enough for equation 43 to give a good guess, but there has not been time to experiment with this idea.

Loose Coupling: Two-Level Newton with Continuation

A very robust approach to obtaining the solution to PDE sub-problems is to apply a simple continuation algorithm to the PDE. At the beginning of each inner loop PDE solve, an assessment is made of how much the connected circuit node voltages have changed. If this change is large, then instead of a single Newton solve, the inner PDE problem is addressed with a series of Newton solves is undertaken. The electrode voltage boundary conditions are changed gradually prior to each solve, and each solve is converged completely before moving on to the next one.

Initially, we implemented the continuation algorithm to vary the electrode potentials by a fixed step-size ΔV at each continuation step. The maximum voltage change to which a PDE device can be subjected and still achieve convergence is problem dependent, so it is difficult to predict exactly how much change can be tolerated by each PDE. Using a constant step-size approach can result in taking a prohibitive number of continuation steps, to account for the worst case scenario.

To overcome this we implemented a simple variable step-size continuation algorithm, in which the step-size ΔV changes depending upon the success or failure of the previous attempted step. Each time a continuation step is successful the step-size is increased by a factor α. For each failure, the step-size is reduced

by another factor β, and the step is re-taken. For this work, we chose $\alpha = 1.5$ and $\beta = 0.125$, but future experiments may determine better numbers. To prevent the scheme from being too aggressive, the step-size is only increased when the number of successive successful steps is equal to or greater than the most recent number of successive failed steps. By using this variable step-size algorithm, the number of required continuation steps was often reduced by over an order of magnitude compared with the constant step-size approach.

This approach to the two-level Newton has proven to be very robust. As of this writing, the inner-loop solve has never failed using this approach, but it has the disadvantage of being relatively slow. If the circuit voltages are changing a great deal, the continuation algorithm may take hundreds of steps, even using the variable step-size algorithm. If the number of continuation steps required for the inner solve is steadily increasing, that is usually a sign that the two-level algorithm is diverging. Fortunately, for most circuits large changes in nodal voltages only occur during the first few outer-loop Newton steps.

Voltage Limiting with Two-Level Newton

Voltage limiting can be applied to mixed-level simulation, and is often neccessary for cases where the circuit portion of the problem is difficult to solve. Applying it in the case of the full-Newton approach is logistically problematic, but applying voltage limiting within the context of a two-level algorithm is straightforward. The two-level Newton is easier because the correction term in equation 36 can be applied in terms of the extracted conductances, instead of in terms of the equations of the PDE device.

Additionally, device simulation usually requires conventional Newton damping, but applying damping to the PDE device part of the problem, while applying voltage limiting to the circuit part of the problem, is very difficult. With a two-level approach, this issue is avoided. Also, by applying voltage limiting on the circuit level, the change in nodal voltages seen by the PDE device is relatively small, and the number of required continuation steps is correspondingly small.

4 Results

A number of different simulation problems are presented in this section and the point of each one is to illustrate a different issue for mixed-level simulation. DC operating point (steady-state) calculations are crucial for circuit simulation as they provide the initial condition for transient and small-signal analysis. Additionally, they are generally the most difficult nonlinear problems to solve as typically they do not have a good initial guess, and unlike transient simulation, there is no obvious analogy to a timestep that may be varried to obtain convergence. For this reason, the results presented here are primarily operating point calculations.

Much of the focus of this work has been on nonlinear solver issues, so to avoid complicating the issue, the same linear solver, SuperLU [9], was used for all problems at all levels of solution.

For all of the examples in this document, the PDE devices are Bipolar Junction Transistors (BJTs). This choice was based on several factors. As of this writing, the PDE modeling capability in **Xyce** does not handle oxide materials, so metal-oxide-semiconductor (MOS) devices were not yet an option. BJTs are a very common type of transistor, used in many circuits at Sandia. Also, they are truly a two-dimensional device, so they are a good test of the two-dimensional capability of **Xyce**. Finally, the most difficult circuits in the **Xyce** test suite, from a nonlinear solver point of view, have usually been BJT circuits.

4.1 Bipolar transistor circuit

A simple circuit schematic, including an NPN bipolar junction transistor (BJT) is shown in Figure 8. The transistor is modeled as a PDE problem. This problem is an example in which the external circuit attached to the device is trivial, while the PDE problem is potentially difficult, depending on the voltages applied to the three device electrodes.

The transistor has three doping regions, as can be seen in Figure 9. Each region corresponds to an electrode, resulting in three electrodes which must be attached to the external circuit. The smallest region, the N+ region, has a doping level of 1.0e+19 donors per cubic centimeter, and is associated with the emitter terminal. The middle P region has a doping level of 1.0e+16 acceptors per cubic centimeter, and is associated with the base terminal. The remaining N region is associated with the collector terminal, and has a doping level of 1.0e+14 acceptors per cubic centimeter.

The mesh for the BJT is in Figure 10, and was generated using the SGFramework [14] mesh generator. It has 1828 mesh points, and has been refined near the boundaries between doping regions, as these tend to be the areas of maximum gradient for all three solution variables. As with the region diagram, the N+, P and N regions are associated with the the emitter, base and collector electrodes, repectively.

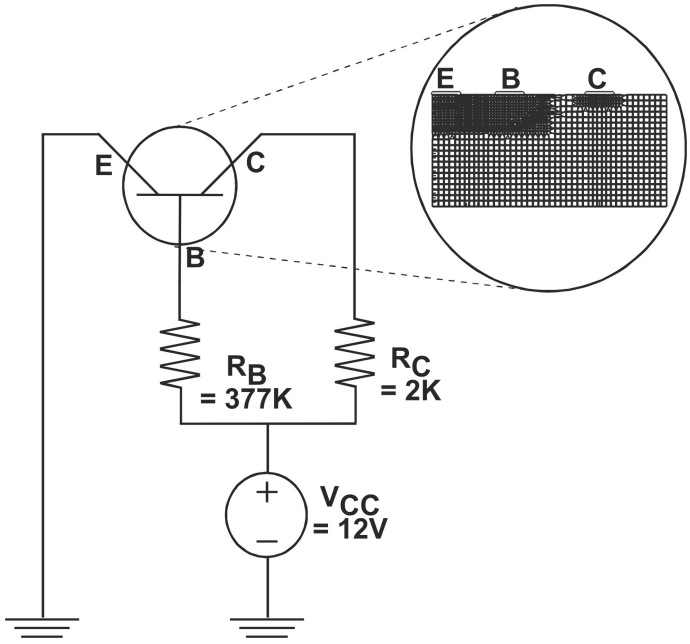

Figure 8. Bipolar Transistor Circuit. E=emitter, B=base, C=collector.

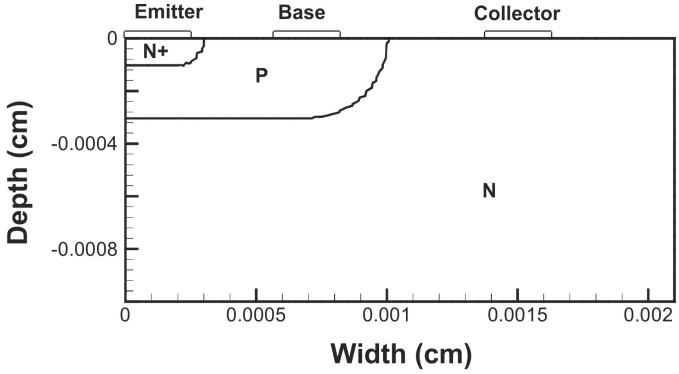

Figure 9. BJT Regions.

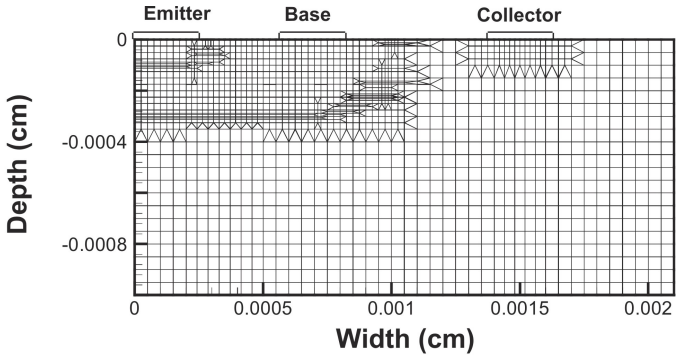

Figure 10. BJT Mesh.

The results of two different operating point calculations are shown in Figures 11 and 12. For the first calculation (Figure 11), the voltage applied by the voltage source, V_{CC} is set to zero, which results the device being completely unbiased. For the second calculation (Figure 12), V_{CC} is set to 12 volts. The application of this voltage results in biases on the base and collector, of approximately 5 and 6 volts respectively, once the calculation has fully converged. The remaining 7 and 6 volts (of the original 12) are dropped across the resistors R_B and R_C.

For the first case, with $V_{CC} = 0V$, the circuit problem and the PDE device problem are reasonably easy to solve. All of the algorithms described in the previous section are successful, provided they begin with a good initial guess. The results are described in the Table 1. For an unbiased problem the nonlinear Poisson initial guess is very close to the converged DD solution, so results with it and without it are included in the table. Problems not including this guess are initialized using

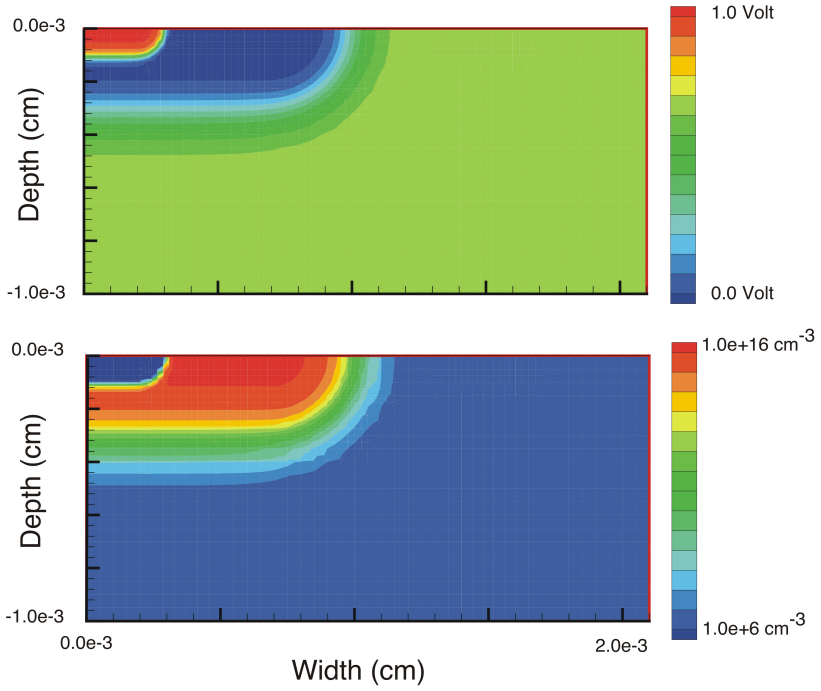

Figure 11. Unbiased BJT Results. The top figure is electrostatic potential, the bottom is hole density.

equations 19- 23. Solving the nonlinear Poisson equation required 6 undamped Newton steps. Interestingly, the full Newton is the most robust approach as it is the only algorithm which achieves convergence without a nonlinear Poisson initial guess. The two level approach requires the solution of several addition linear systems, so the total solve time is longer. The full Newton approach is clearly preferable for a problem as easy as this one, but this is not the case in general.

Method	**Outer Newton Steps**	**Inner Newton Steps**	**Continuation Steps**
Full Newton	15	NA	NA
Two-Level Newton	-	-	NA
Two-Level Newton, with Continuation	-	-	NA
Full Newton, w/nonlinear Poisson initial guess	1	NA	NA
Two-Level Newton w/nonlinear Poisson initial guess	1	1	NA
Two-Level Newton, w/Continuation and nonlinear Poisson initial guess	1	1	1

Table 1: Nonlinear solver results for unbiased BJT circuit. The entry ”-” indicates that the problem did not converge.

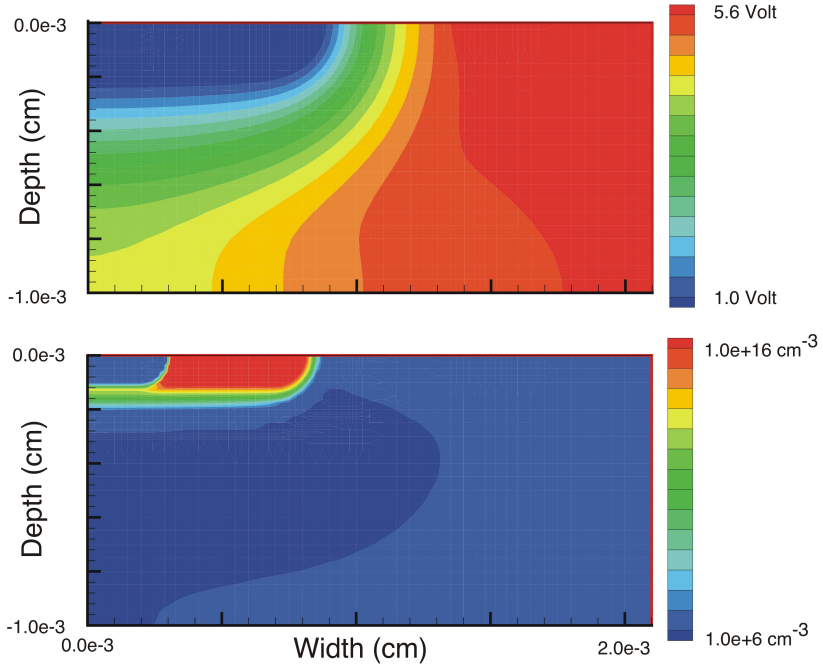

Figure 12. Biased BJT Results. The top figure is electrostatic potential, the bottom is hole density.

For the second case, with $V_{CC} = 12V$, the external circuit portion of the problem remains easy to solve. The PDE device problem, however, is more difficult because now a large bias is applied to the base and collector. The results of the second calculation are in Figure 12. The electrostatic potential has a much more dramatic variation than in the unbiased case. Also, note that the areas of high gradient for all variables no longer correspond closely with the doping region boundaries. As a result, predicting a good initial guess is more difficult.

The nonlinear solver performance results are described in Table 2. This problem proved difficult for the nonlinear solver, as only two algorithms were successful. Both of the successful approaches used the two-level Newton with continuation, and even that approach required a good initial guess for the PDE problem. The large number of steps resulted in a relatively long total solution time. This solution was approximately 300 times slower (about 45 minutes compared with 8 seconds on a particular workstation) for full Newton solve of the unbiased case. Applying voltage limiting to the circuit phase of the solve improved the performance substantially, bringing it down to about 10 minutes. (note that these wall clock times were not performed with optimized code)

One aspect of the two-level Newton approach is that extra diagnostics are avail-

able for estimating the real-time success of the solve. As the two-level algorithm progressed, the number of required inner loop continuation steps decreased. For some of the early outer Newton loop steps, the continuation loop took 50-60 steps. By the end of the outer loop, the continuation loop was down to a single step.

Method	**Outer Newton Steps**	**Inner Steps**	**Continuation Steps**
Full Newton	-	NA	NA
Two-Level Newton	-	-	NA
Two-Level Newton, with Continuation	-	-	NA
Full Newton, w/nonlinear Poisson initial guess	-	NA	NA
Two-Level Newton w/nonlinear Poisson initial guess	-	-	NA
Two-Level Newton, w/Continuation and nonlinear Poisson initial guess	56	2470	447
Two-Level Newton, w/Continuation, nonlinear Poisson and voltage limiting	18	698	112

Table 2: Nonlinear solver results for biased BJT circuit. The entry ”-” indicates that the problem did not converge.

Changing the value of V_{CC} from $0V$ to $12V$ resulted in a significant difference in solver performance. Generally, for operating point problems, one of the challenges for mixed-level simulation is the application of nonzero operating point voltage sources to the PDE sub-problems. This is one of the main issues that can cause the PDE sub-problem to be hard to solve.

4.2 Bipolar Inverter Circuit

Figure 13 shows a schematic diagram for a multiple PDE-device circuit problem. This circuit is a digital inverter, in which the state of the output node should be the opposite that of the input node. A circuit like this is good for scaling studies, as it can be viewed as a unit cell, of which an arbitrary number can be chained together. There are two PDE bipolar transistors in Figure 13, and they both use the same mesh as the transistor in the previous example. One of the two transistors is a PNP, and has the inverse doping of the NPN.

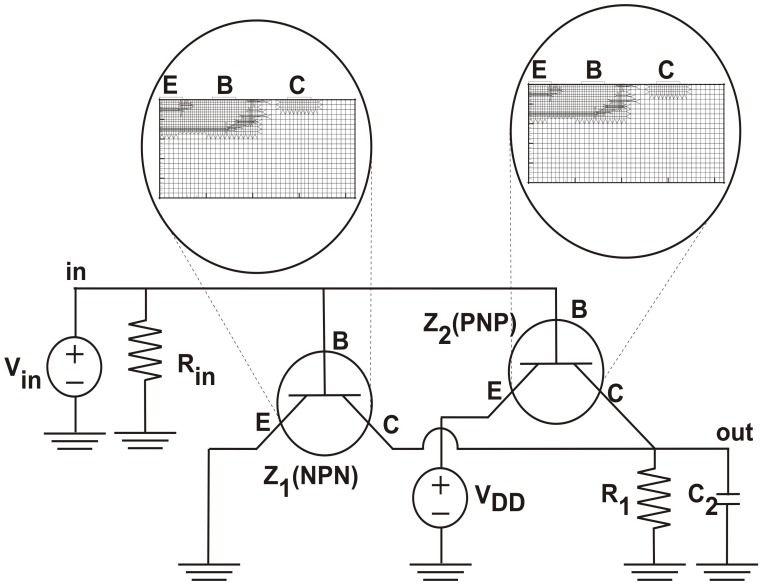

Figure 13. Inverter Schematic.

The result for a transient simulation of the inverter is shown in Figure 14. Nonlinear solver results are shown in Table 3. For all the results presented here, the nonlinear Poisson solution was used as an initial guess, as without it, none of the methods would have been successful. As in the previous example, the only successful algorithms used the two-level Newton with continuation, and applying voltage limiting to the circuit phase significantly improved the solver performance.

Several scaling tests were performed with the inverter, using the circuit shown in Figure 13 as a unit cell. DC operating point calculations were performed for inverter chains with 2, 4, 6, and 8 BJT PDE devices. Interestingly, but perhaps not suprisingly, the nonlinear solver performance for each case was nearly identical as in the 2 PDE case. For each case, the number of required outer Newton steps was 7, when the two-level Newton with continuation and voltage limiting was used.

The implementation of the two-level algorithm in **Xyce** solves all PDE devices simultaneously, as part of the same inner-loop Newton solve. (See Figure 7.) Solving all the inner problems simultaneously in this manner did not appear to cause any problems, as each PDE device is completely isolated. However, this implementation can result in the solver having to do unneccessary work, as it is possible to have one PDE device that is more difficult to solve than all the others, and the Newton solver will continue to iterate on all sub-problems until all of them are converged.

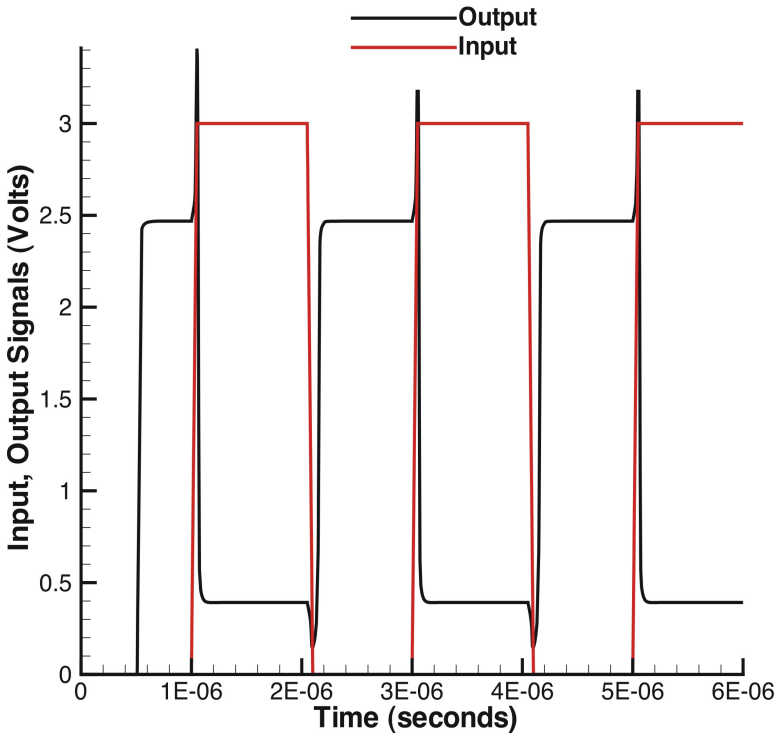

Figure 14. Inverter Circuit Result.

The result for a transient simulation of the inverter is shown in Figure 14. The transient simulation used the two-level-Newton approach with continuation to obtain the DC operating point, and then switched to full Newton for the transient phase. This is an easy result to verify, at least to first order. The input signal should be the opposite of the output signal. When one is high, the other should be low, and vice-versa.

The results presented in Table 3 notwithstanding, it is often possible to obtain convergence with the full-Newton algorithm for mixed-level problems, even with high-valued voltage sources. However, doing so requires that the user apply a continuation algorithm by hand, either by using the DC sweep capability of **Xyce**, or, in the case of a transient simulation, gradually ramping up the voltage source(s) over time, and letting the time integrator handle step size control, rather than the continuation algorithm.

Using a manual approach such as this can work reasonably well for simple problems, but it requires some sophistication on the part of the user. Unfortunately, the successful use of such an approach requires that the optimal nonlinear solver settings for the circuit problem be similar to those for the PDE problem. This is usu-

ally not the case in practice. A typical problem, for which the sub-problem solver requirements are significantely different, is presented in the next example.

Method	**Outer Newton Steps**	**Inner Newton Steps**	**Continuation Steps**
Full Newton	-	NA	NA
Two-Level Newton	-	-	NA
Two-Level Newton, w/Continuation	16	358	83
Two-Level Newton, w/Continuation and voltage limilting	7	276	49

Table 3: Nonlinear solver results for BJT inverter circuit. The entry ”-” indicates that the problem did not converge.

4.3 RCA3040 Wideband Amplifier

The circuit portion of the previous examples has been relatively easy to solve. In both cases, the few non-PDE devices are linear resistors, capacitors, and voltage sources. A more challenging test of a coupling algorithm is one in which both the circuit problem and the PDE problem are difficult. A number of circuits in the **Xyce** test suite only successfully converge when voltage limiting is applied. One such circuit is the RCA3040 amplifier, which is from the CircuitSim90 set of benchmarks [1]. This circuit consists of 11 BJTs, 12 resistors, and three voltage sources. For the operating point calculation, one voltage source is set to 15 volts, one to -15 volts, and the other to zero.

As a mixed-level test for **Xyce**, one of the BJTs was replaced by a PDE device. The mesh was slightly courser than in the previous two examples, but otherwise the device was the same, with the same doping levels and geometry. The results of this test are shown in table 4. The only successful algorithm for this case is the final one, in which a two-level Newton with continuation is used, with voltage limiting applied to the circuit phase. All other algorithms failed.

This illustrates a big advantage of the two-level Newton, that the circuit and PDE device phases of the problem are separate, and can have different algorithms applied. Voltage limiting is a nonlinear solver technique that is unique to circuit simulation. In most implementations (including that of **Xyce**) it is incompatible with nonlinear solver algorithms which depend upon the solution history. As such, conventional Newton damping, which is generally the best algorithm to use for PDE device simulation, is incompatible with voltage limiting, the most effective technique for circuits.

As mentioned earlier, voltage limiting has long been recognized as problematic, and a lot of modern circuit codes no longer use it, preferring homotopy methods [18, 19] instead. Currently, homotopy methods are not availble in **Xyce**, but they are a planned enhancement. The advantages of using the two-level Newton will remain, however, as it may prove difficult to design homotopy algorithms that work effectively for the full problem.

Method	**Outer Newton Steps**	**Inner Newton Steps**	**Continuation Steps**
Full Newton	-	NA	NA
Two-Level Newton	-	-	NA
Two-Level Newton, w/Continuation	-	-	-
Two-Level Newton, w/Continuation and voltage limiting	7	311	44

Table 4: Nonlinear solver results for RCA3040 amplifier circuit. The entry ”-” indicates that the problem did not converge.

5 Conclusions and Future Work

The subject of this work has been mixed-level simulation. Two PDE devices, a one dimensional and a two dimensional device, have been implemented within the **Xyce** circuit simulator. A variety of mixed-level algorithms have been implemented and tested, including full-Newton, two-level Newton, modified two-level Newton, two-level Newton with continuation. Additionally, voltage limiting on the circuit level has been expanded to work with two-level Newton. Based on preliminary tests, the most robust of these approaches appears to be two-level Newton with continuation and voltage limiting.

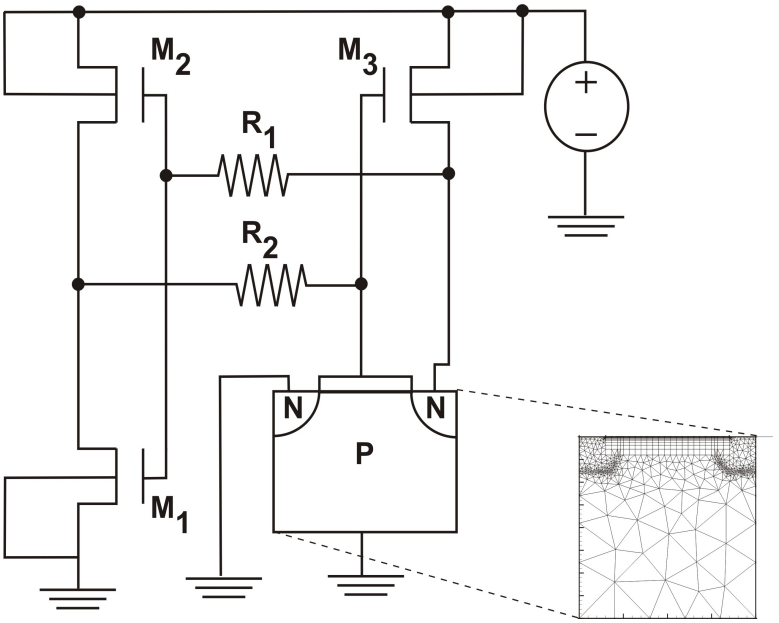

Figure 15. Schematic for single-event-upset (SEU).

The two-level approaches were generally more robust than the full Newton, because using it allows for more flexibility. Circuits and PDE devices generally have different solver requirements, and by using a two-level approach, there is no need to compromise. Solver settings that are best for the circuit can be applied to the outer loop, while the best settings for PDE devices can be applied at the inner loop.

Future work will include further code development. One simulation problem that requires mixed-level simulation is Single Event Upset (SEU), and this problem is of interest to Sandia. (see Figure 15) The PDE devices in **Xyce** currently do not support oxide materials, which is neccessary for problems like SEU which include

MOS (metal-oxide-semiconductor) devices. Future releases of **Xyce** will include oxide material support.

Additionally, the devices that are built directly into **Xyce** are of at most two dimensions. A new project, **Charon** has recently started to develop a three dimensional device modeling capability. This new capability will allow for the device to be partitioned in parallel over multiple processors. The PDE devices in **Xyce** are assumed to be self-contained on a single processor, which is reasonable given their typical size. As **Charon** is designed to use many of the same solver libraries as **Xyce**, it should be possible to take advantage of many of the lessons learned in this project, and apply them to linking **Xyce** and **Charon** together.

Future work on coupling algorithms will include investigation of the modified two-level Newton, which we attempted, but not successfully. Possibly combining the modified two-level Newton with the right degree of voltage limiting at the circuit level will result in a robust algorithm.

Finally, the success of continuum methods, when applied to the PDE problem, points in the direction of homotopy methods in general. Voltage limiting has long been used in circuit simulation, but it has a number of drawbacks, including incompatibility with other nonlinear methods. In recent years, modern circuit codes have moved away from using voltage limiting and have applied homotopy methods with much success [18, 19]. The application of homotopy methods in **Xyce** is expected in the coming year, to take advantage of the **LOCA** [4] library that has been added to **NOX**. How to best incorporate circuit-level homotopy in the context of the two-level Newton is a subject for future investigation.

References

[1] *CircuitSim90 Benchmark Information*. `http://www.cbl.ncsu.edu/benchmarks/`, 1990.

[2] Hspice user's manual. Technical report, Meta-Software, Inc., Campbell, California, 1996.

[3] Davinci user's manual. Technical report, Avanti! Corporation TCAD Business Unit, 1998.

[4] *LOCA: Library of Continuation Algorithms*. `http://www.cs.sandia.gov/projects/loca/main.html`, 2002.

[5] *The Trilinos Project*. `http://www.cs.sandia.gov/Trilinos/`, 2002.

[6] P. Antognetti and G. Massobrio. *Semiconductor Device Modeling with SPICE*. McGraw-Hill, 1988.

[7] Achim Basserman. Private communication. 2002.

[8] Leon O. Chua and Pen-Min Lin. *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*. Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

[9] James W. Demmel, John R. Gilbert, and Xiaoye Li. *SuperLU Users' Guide*. `http://www.nersc.gov/ xiaoye/SuperLU/`, 1999.

[10] H. K. Dirks and K. Eickhoff. Numerical models and table models for mos circuit analysis. *Proc. NASECODE IV*, pages 13–24, June 1985.

[11] Bruce Hendrickson and Robert Leland. The Chaco User's Guide: Version 2.0. Technical Report SAND94–2692, Sandia National Laboratories, Albuquerque, NM, December 1994.

[12] Scott Hutchinson. Xyce users' guide. Technical report SAND98-xxxx, Sandia National Laboratories, Albuquerque, New Mexico 87185, 2002.

[13] Eric R. Keiter. Xyce parallel electronic simulator design: Mathematical formulation. Technical report SAND02-xxxx, Sandia National Laboratories, Albuquerque, New Mexico 87185, 2002.

[14] Kevin M. Kramer and W. Nicholas G. Hitchon. *Semiconductor Devices: A Simulation Approach*. Prentice-Hall, Upper Saddle River, New Jersey, 1997.

[15] Kartikeya Mayaram and Donald O. Pederson. Codecs: A mixed-level device and circuit simulator. *Proc. IEEE Int. Conf. Computer-Aided Design*, pages 112–115, November 1988.

[16] Kartikeya Mayaram and Donald O. Pederson. Coupling algorithms for mixed-level circuit and device simulation. *IEEE Transactions on Computer Aided Design*, II(8):1003–1012, 1992.

[17] J.R.F. McMacken and S.G.Chamberlain. Chord: A modular semiconductor device simulation tool incorporating external network models. *IEEE Transactions on Computer Aided Design*, 8:826–836, August 1989.

[18] Robert C. Melville, Ljiljana Trajkovic, San-Chin Fang, and Layne T. Watson. Globally convergent homotopy methods for the dc operating point problem. Technical report TR 90-61, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061.

[19] Robert C. Melville, Ljiljana Trajkovic, San-Chin Fang, and Layne T. Watson. Artificial parameter homotopy methods for the dc operating point problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(6):861–877, 1993.

[20] M. S. Mock. Time-dependent simulation of coupled devices. *Proc. NASEC-ODE II*, pages 113–131, June 1981.

[21] Laurence Nagel and Ronald Rohrer. Computer analysis of nonlinear circuits, excluding radiation (cancer). *IEEE Journal of Solid-State Circuits*, sc-6(4):166–182, 1971.

[22] Tom Quarles. Spice 3 version 3f5 users' manual. *University of California, Berkeley, CA*, 1994.

[23] Tom Quarles. Spice3f5 users' guide. Technical report, University of California-Berkeley, Berkeley, California, 1994.

[24] Gregory J. Rollins and John Choma. Mixed-mode pisces-spice coupled circuit and device solver. *IEEE Transactions on Computer Aided Design*, 7:862–867, August 1988.

[25] Francis Rotella. Mixed circuit and device simulation for analysis, design, and optimization of opto-electronic, radio requency, and high speed semiconductor devices. Phd thesis, Stanford University, 2000.

[26] C. W. Ho A. E. Ruehli and P. A. Brennan. The modified nodal approach to network analysis. *IEEE Trans. Circuits Systems*, 22:505–509, 1988.

[27] D. L. Scharfetter and J. K. Gummel. Large-signal analysis of a silicon read diode oscillator. *IEEE Trans. on Electron Devices*, ED-16:64–77, 1969.

[28] S. Selberherr. *Analysis and Simulation of Semiconductor Devices*. Springer-Verlag, New York, 1984.

[29] Jiri Vlach and Kishore Singal. *Computer Methods for Circuit Analysis and Design*. Chapman and Hall, New York, 1994.

[30] R. Laur W. L. Engl and H. K. Dirks. Medusa - a simulator for modular circuits. *IEEE Trans. Computer-Aided Design*, CAD-1:85–93, 1982.

[31] Tor A. Fjeldly Trond Ytterdal and Michael Shur. *Introduction to Device Modeling and Circuit Simulation*. John Wiley and Sons, New York, 1988.

[32] Z. Yu, D. CHen, L. So, , and R. W. Dutton. Pisces-2et–two dimensional device simulation for silicon and heterostructures. Technical report, Stanford University, 1994.

DISTRIBUTION:

1 Steven P. Castillo
Klipsch School of Electrical and Computer Engineering
New Mexico State University
Box 3-o
Las Cruces, NM 88003

1 Kwong T. Ng
Klipsch School of Electrical and Computer Engineering
New Mexico State University
Box 3-o
Las Cruces, NM 88003

1 Nick Hitchon
Electrical and Computer Engineering
University of Wisconsin
1415 Engineering Drive
Madison, WI 53706

1 Deborah Fixel
Electrical and Computer Engineering
University of Wisconsin
1415 Engineering Drive
Madison, WI 53706

1 Mark Kushner
Department of Electrical and Computer Engineering
University of Illinois
1406 W. Green Street
Urbana, IL 61801

1 Andrew J. Christlieb
Department of Mathematics
University of Michigan
2470 East Hall
Ann Arbor, MI 48109

1 Ron Kielkowski
RCG Research, Inc
8605 Allisonville Rd, Suite 370
Indianapolis, In 46250

1 Mike Davis
Software Federation, Inc.
211 Highview Drive
Boulder, Co 80304

1 Wendland Beezhold
Idaho Accelerator Center
1500 Alvin Ricken Drive
Pocatello, Idaho 83201

1 Kartikeya Mayaram
Department of Electrical and Computer Engineering
Oregon State University
Corvallis, OR 97331-3211

1 Linda Petzold
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106-5070

1 Jaijeet Roychowdhury
4-174 EE/CSci Building
200 Union Street S.E.
University of Minnesota
Minneapolis, MN 55455

1 C.-J. Richard Shi
VLSI and Electronic Design Automation
210 EE/CSE Bldg.
Box 352500
University of Washington
Seattle, WA 98195

1 Homer F. Walker
WPI Mathematical Sciences
100 Institute Road
Worcester, MA 01609

1 Dan Yergeau
CISX 334
Via Ortega
Stanford, CA 94305-4075

1 Masha Sosonkina
319 Heller Hall
10 University Dr.
Duluth, MN 55812

1 Misha Elena Kilmer
113 Bromfield-Pearson Bldg.
Tufts University
Medford, MA 02155

1 Tim Davis
P.O. Box 116120
University of Florida
Gainesville, FL 32611-6120

1 Achim Basermann
C&C Research Laboratories,
NEC Europe Ltd.
Rathausallee 10
D-53757 Sankt Augustin
Germany

1 Philip A. Wilsey
Experimental Computing Laboratory
Department of Electrical & Computer Engineering and Computer Science
College of Engineering
P.O. Box 210030
University of Cincinnati
Cincinnati, Ohio 45221-0030

1 Dale E. Martin
Clifton Labs
3678 Fawnrun Dr.
Cincinnati, OH 45241

1 Vladimir Kolobov
CFD Research Corporation
215 Wynn Drive Huntsville, AL 35805

1 Richard L. Keiter
Department of Chemistry
Eastern Illinois University
600 Lincoln Avenue
Charleston, IL 61920

1 Ellen A. Keiter
Department of Chemistry
Eastern Illinois University
600 Lincoln Avenue
Charleston, IL 61920

1 Lise Keiter-Brotzman
10 College Circle
Staunton, VA 24401

1 Al Lehnen
Mathematics Department
Madison Area Technical College
3550 Anderson Street
Madison, WI 53704

1 MS 0321
Bill Camp, 09200

1 MS 0318
Paul Yarrington, 09230

1 MS 1071
Mike Knoll, 01730

1 MS 0310
Robert Leland, 09220

1 MS 0316
Sudip Dosanjh, 09233

1 MS 0525
Paul V. Plunkett, 01734

1 MS 0835
J. Michael McGlaun, 09140

1 MS 0139
Stephen E. Lott, 09905

1 MS 0310
Mark D. Rintoul, 09212

1 MS 1110
David Womble, 09214

1 MS 1111
Bruce Hendrickson, 09215

1 MS 0819
Edward Boucheron, 09231

1 MS 0819
Allen C. Robinson, 09231

1 MS 0316
John Aidun, 09235

1 MS 0316
Scott A. Hutchinson, 09233

10 MS 0316
Eric R. Keiter, 09233

1 MS 0316
Robert J. Hoekstra, 09233

1 MS 0316
Joseph P. Castro, 09233

1 MS 0316
David R. Gardner, 09233

1 MS 0316
Gary Hennigan, 09233

1 MS 0316
Roger Pawlowski, 09233

1 MS 0316
Richard Schiek, 09233

1 MS 1111
John N. Shadid, 09233

1 MS 1111
Andrew Salinger, 09233

1 MS 0316
Paul Lin, 09233

1 MS 0847
Scott Mitchell, 09211

1 MS 0847
Mike Eldred, 09211

1 MS 0847
Bart van Bloemen Waanders, 09211

1 MS 0847
Roscoe A. Bartlett, 09211

1 MS 0196
Elebeoba May, 09212

1 MS 1110
Todd Coffey, 09214

1 MS 1110
David Day, 09214

1 MS 1110
Mike Heroux, 09214

1 MS 1110
Pavel B. Bochev, 09214

1 MS 1110
Richard B. Lehoucq, 09214

1 MS 1110
Raymond S. Tuminaro, 09214

1 MS 1110
James Willenbring, 09214

1 MS 1111
Karen Devine, 09215

1 MS 1109
Robert Benner, 09224

1 MS 0316
Harry Hjalmarson, 09235

1 MS 0525
Steven D. Wix, 01734

1 MS 0525
Thomas V. Russo, 01734

1 MS 0525
Lon Waters, 01734

1 MS 0525
Regina Schells, 01734

1 MS 0525
Carolyn Bogdan, 01734

1 MS 0525
Mike Deveney, 01734

1 MS 0525
Raymond B. Heath, 01734

1 MS 0525
Ronald Sikorksi, 01734

1 MS 0525
Albert Nunez, 01734

1 MS 1081
Paul E. Dodd, 01762

1 MS 0660
Roger F. Billau, 09519

1 MS 0874
Robert Brocato, 01751

1 MS 1081
Charles E. Hembree, 01739

1 MS 0311
Greg Lyons, 02616

1 MS 0311
Martin Stevenson, 02616

1 MS 0328
Fred Anderson, 02612

1 MS 0537
Perry Molley, 02331

1 MS 0537
Siviengxay Limary, 02331

1 MS 0537
John Dye, 02331

1 MS 0537
Barbara Wampler, 02331

1 MS 0537
Doug Weiss, 02333

1 MS 0537
Scott Holswade, 02333

1 MS 0481
Joel Brown, 02132

1 MS 0405
Todd R. Jones, 12333

1 MS 0405
Thomas D. Brown, 12333

1 MS 0405
Donald C. Evans, 12333

1 MS 9101
Rex Eastin, 08232

1 MS 9101
Seung Choi, 08235

1 MS 9409
William P. Ballard, 08730

1 MS 9202
Kathryn R. Hughes, 08205

1 MS 9202
Rene L. Bierbaum, 08205

1 MS 9202
Kenneth D. Marx, 08205

1 MS 9202
Stephen L. Brandon, 08205

1 MS 9217
Stephen W. Thomas, 08950

1 MS 9217
Tamara G. Kolda, 08950

1 MS 9217
Kevin R. Long, 08950

1 MS 1179
Leonard Lorence, 15341

1 MS 1179
David E. Beutler, 15341

1 MS 1179
Brian Franke, 15341

1 MS 0835
Randy Lorber, 09141

1 MS 1152
Mark L. Kiefer, 01642

1 MS 9018
Central Technical Files, 8945-1

2 MS 0899
Technical Library, 9616

1 MS 0612
Review & Approval Desk, for DOE/OSTI, 9612